Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

J. Math. Anal. Appl. ••• (••••) •••–•••

Journal of MATHEMATICAL ANALYSIS AND APPLICATIONS

www.elsevier.com/locate/jmaa

A derivative-free optimization algorithm based on conditional moments ☆

Xiaogang Wang, Dong Liang *, Xingdong Feng, Lu Ye

Department of Mathematics and Statistics, York University, Toronto, Ontario, Canada M3J 1P3

Received 10 June 2006

Submitted by Steven G. Krantz

Abstract

In this paper we propose a derivative-free optimization algorithm based on conditional moments for finding the maximizer of an objective function. The proposed algorithm does not require calculation or approximation of any order derivative of the objective function. The step size in iteration is determined adaptively according to the local geometrical feature of the objective function and a pre-specified quantity representing the desired precision. The theoretical properties including convergence of the method are presented. Numerical experiments comparing with the Newton, Quasi-Newton and trust region methods are given to illustrate the effectiveness of the algorithm.

Keywords: Optimization; Derivative-free; Conditional moment; Trust region

1. Introduction

Optimization has been playing an important role in many branches of science and technology such as engineering, finance, probability and statistics (see, for example, [7,10,11,20], etc.). There are many optimization algorithms that have been developed to locate the optima of

☆ This work was supported by National Engineering and Science Research Council of Canada.
* Corresponding author.
E-mail address: dliang@mathstat.yorku.ca (D. Liang).

0022-247X/$ – see front matter
doi:10.1016/j.jmaa.2006.08.091

Please cite this article as: X. Wang et al., A derivative-free optimization algorithm based on conditional moments, J. Math. Anal. Appl. (2006), doi:10.1016/j.jmaa.2006.08.091

continuous objective functions. We are concerned with the maximization problem of a smooth function f of several variables. Formally, we seek the solution of the following problem:

$$\max_{\mathbf{x}\in\mathbf{D}} f(\mathbf{x}), \tag{1}$$

where $\mathbf{D} \subset \mathbf{R^n}$ is a bounded domain.

For an optimization problem with no constraint, one widely used method is the Newton method. For the Newton–Raphson algorithm, the iteration is defined by

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_i G_i^{-1} \mathbf{g}_i,$$

where α_i is the step size, $\mathbf{g}_i$ is the gradient vector, and G_i is the Hessian matrix. The Newton method converges fast in general and could be very efficient for smooth objective functions. However, the Hessian matrix is rather difficult or impossible to obtain in many practical problems. To rectify this problem, the Quasi-Newton algorithm is proposed by Davidon [5]. The basic idea of the Quasi-Newton algorithm is to use an iterative matrix H_i to approximate Hessian matrix G_i^{-1}. The BFGS Quasi-Newton method by Broyden [2]–Fletcher [6]–Goldfarb [8]–Shanno [17] has been proved reliable and efficient for the unconstrained minimization of a smooth function. Although the matrices H_i's are positive definite in theory, it is well known that they are often not the case especially for high-dimensional space due to rounding errors. Also, ill-conditioned matrices can cause serious numerical problems in practice. There are many modified versions of Newton method such as the Damped Newton method and the *Lavenberg–Marquardt* type method (see [7,13]). A polynomial-time algorithm for linear problems has also been proposed (see [4]). However, all these methods must require the calculation of the derivatives of objective functions.

Another well-known problem with Newton method and the modified methods based on the Newton method is that the initial values are usually required to lie within a relatively small neighbourhood of the true optimum to ensure any desired accuracy. For example, one of the most widely used algorithms in statistics is the so-called Expectation–Maximization (EM) algorithm. It has been observed that this algorithm converges slowly and is very sensitive to the initial value. The problem lies in the maximization step of the algorithm in which the Newton or Quasi-Newton methods are employed (see [9]). Furthermore, some objective functions encountered in practice could be either very flat or have quite large first-order derivative near the global optima. This creates additional challenges to the Newton or Quasi-Newton type of algorithms as the accurate evaluation of the derivatives and the choice of step size are crucial.

On the other aspect, the derivatives of objective functions might not be available in many applications of maximization. Therefore, there have been considerable interests in developing effective algorithms that are of derivative-free. The trust region methods are widely studied and successful in the literature (see, for example, [12,15,19], etc.). For example, one of very recent trust region methods is called wedge trust region method (see Marazzi and Nocedal [12]). The wedge trust region method employs a model which interpolates the objective function at a set of sample points. The model is built upon the trust region framework such that the convergence of the model is guaranteed. Therefore, the model in wedge trust region method can be in either linear or quadratic order.

In this paper, we propose a novel derivative-free algorithm for the general optimization problem based on conditional moments. The proposed algorithm is built upon a direct evaluation of the conditional moments of a non-negative function which represents the local centre of gravity of a mass function. The algorithm constructs a path defined by the geometric centres of the objective function within a series small neighbourhoods so that it will travel dynamically towards the global optimum. The proposed algorithm is free of any order derivatives of the objective

Please cite this article as: X. Wang et al., A derivative-free optimization algorithm based on conditional moments, J. Math. Anal. Appl. (2006), doi:10.1016/j.jmaa.2006.08.091

functions and only depends on local integrations which are evaluated by the numerical quadratures such as the composite Simpson quadratures and the composite quadratures of Gaussian types. There are two iterative parameters in the procedure and they will be valued adaptively in the algorithm. For the proposed method, we have further established the theoretical properties including its convergence. Numerical experiments comparing with the Newton, Quasi-Newton and trust region methods are taken to illustrate the performance of the algorithm. It shows that the algorithm is very effective when the objective functions rise either very sharply or slowly near the global optimum. It is also very effective when there exist multiple local optima with a close proximity of global one.

The remaining part of the paper is organized as follows. In Section 2, we give a general description of the algorithm and we also describe the method of choosing the parameters adaptively. Theoretical properties of the proposed method are established in Section 3. Numerical results comparing the proposed algorithm with the Newton, Quasi-Newton and wedge trust region method are provided in Section 4. Finally, the conclusion is given in Section 5.

2. The derivative-free conditional moment algorithm

2.1. The basic idea

The idea of our derivative-free algorithm originated from the property of conditional moments. The moments are defined in probability (see [1]), which can describe the local centre of gravity of a mass function. For simplicity, we assume that the objective function is positive everywhere. Then the objective function could be considered as a mass density function. Higher value of the objective function would provide more weight in its neighbourhood. For example, consider a symmetric objective function with the only optimum located at 0. The centre of function coincides with the centre of gravity located at 0. In this simple example, finding the optimum is equivalent to finding the centre of gravity. However, for a non-symmetric non-negative functions, these two problems are no longer equivalent as the location of the global optimum often differs from the centre of the gravity. Thus instead of considering the centre of gravity for the entire objective function, we consider the local centre of gravity given a small neighbourhood. This corresponds to the first-order *conditional* moment if the objective function is non-negative. Furthermore, we propose to move dynamically through a series of varying neighbourhoods and the movement is governed by a sequence of local gravity centres of these connected small local regions. Let $\mathbf{x}_0$ represent the centre and $\mathbf{cg}$ represent the centre of gravity in this neighbourhood. It is defined numerically as

$$\mathbf{cg} = \frac{\int \mathbf{x} G(\mathbf{x})\, d\mathbf{x}}{\int G(\mathbf{x})\, d\mathbf{x}}, \tag{2}$$

where $G(\mathbf{x})$ is the non-negative mass density function. The centre of gravity defined by Eq. (2) coincides with the conditional first-order moment if the objective function is non-negative and can be normalized to 1. It can be seen that $G(\mathbf{cg}) > G(\mathbf{x}_0)$ which will be proved formally in Section 3. Then, we can "relocate" $\mathbf{x}_0$ to $\mathbf{cg}$ which occupies a higher "ground" within which the average of objective function is bigger than that of the previous neighbourhood. This action is repeated until the required convergence is achieved.

This motivates us to define a derivative-free optimization method for solving the optimization problems. The detailed description of our algorithm will be given in the following subsections.

Please cite this article as: X. Wang et al., A derivative-free optimization algorithm based on conditional moments, J. Math. Anal. Appl. (2006), doi:10.1016/j.jmaa.2006.08.091

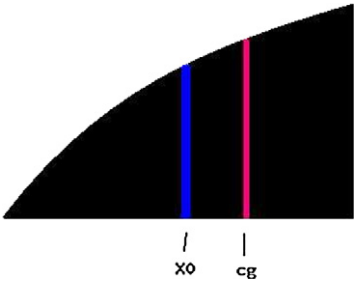

Fig. 1. The local centre of gravity in a monotone neighbourhood.

2.2. The method

In general, any real function $F(\mathbf{x})$ can be decomposed into $F^+(\mathbf{x}) = \max(F(\mathbf{x}), 0)$ and $F^-(\mathbf{x}) = \min(F(\mathbf{x}), 0)$. The positive part $G(\mathbf{x}) = F^+(\mathbf{x})$ is of interest for maximization problems. Thus we focus on non-negative functions in this article. If $F^+(\mathbf{x}) = 0$ for all $\mathbf{x}$, we then set F by $F - C$ where C is a constant such that $G(\mathbf{x}) > 0$ for some $\mathbf{x}$.

Let the objective function $G(\mathbf{x})$, $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in R^n$, be a non-negative continuous function, where n is the dimensionality. Then, for any given α, $\bar{\mathbf{x}} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$ and $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in R^n$, define a set

$$B\big(\bar{\mathbf{x}}, \mathbf{d}(\alpha)\big) = \big\{\mathbf{x} \in R^n\colon\ \|\mathbf{x} - \bar{\mathbf{x}}\| < d(\alpha)\big\} \subset R^n \tag{3}$$

such that

$$\int\limits_{B(\bar{\mathbf{x}}, d(\alpha))} G(\mathbf{y})\,\mathbf{dy} = \alpha, \tag{4}$$

where d of the set $B(\bar{\mathbf{x}}, d)$ depends on the value of α. If α is set to 0, then $B(\bar{\mathbf{x}}, d)$ is a singleton set. To avoid this trivial case, we assume that α_k is positive in the sequel. It is obvious that the value of α is bounded by $\alpha^M = \int_{R^n} G(\mathbf{x})\,d\mathbf{x} > 0$.

Then, we can propose the conditional moment method as follows: From the previous approximation $\mathbf{x}^k$ with parameter $\alpha_k > 0$ and radius parameter $d_k > 0$ obtained from (4) with $\bar{\mathbf{x}} = \mathbf{x}^k$, the new step approximation of the optima is defined as

$$\mathbf{x}^{k+1} = T\big(\mathbf{x}^k, d^k(\alpha_k)\big) = \frac{1}{\alpha_k} \int\limits_{B(\mathbf{x}^k,\, d^k)} \mathbf{y} G(\mathbf{y})\,d\mathbf{y}, \tag{5}$$

with initial guess $\mathbf{x}^0$ being given. The new position $\mathbf{x}^{k+1}$ is the ratio of the first-order moment over the zero-order moment on the local region. It is clear that the iteration at each step $k > 0$ only depends on the local integration over $B(\mathbf{x}^k, d^k)$. If $G(\mathbf{x})$ is a probability density function, then $\mathbf{x}^{k+1}$ represents the conditional mean on $B(\mathbf{x}^k, d^k)$. Meanwhile, the new approximation $\mathbf{x}^{k+1}$ also depends the choice of the parameters α and d.

2.3. The algorithm

There is only parameter α_k in the proposed method which will be selected dynamically. However, the method requires integration over a local region such that Eq. (4) is satisfied. In practice,

Please cite this article as: X. Wang et al., A derivative-free optimization algorithm based on conditional moments, J. Math. Anal. Appl. (2006), doi:10.1016/j.jmaa.2006.08.091

the computational cost to determine such an area for high-dimensional objective function could be prohibitively high. One logical way of realistically handling integration on the set $B(\mathbf{x}^{\mathbf{k}}, d^k)$ is to apply Eq. (5) for each dimension individually while the rest of the coordinates are fixed. The coordinates of the location at the kth iteration, say, $\mathbf{x}^k$ will then be updated one at the time in the same fashion of the well-known Gibbs sampler in physics and statistics (see [16]).

Therefore, we then seek to find $\mathbf{r}^{\mathbf{k}} = (r_1^k, r_2^k, \ldots, r_n^k)$, $i = 1, 2, \ldots, n$, such that

$$\eta\left(\mathbf{x}^k, \mathbf{r}^k\right) = \int_{x_1^k - r_1^k}^{x_1^k + r_1^k} \cdots \int_{x_n^k - r_n^k}^{x_n^k + r_n^k} G(x_1, x_2, \ldots, x_n)\, dx_1 dx_2 \cdots dx_n - \alpha_k = 0. \tag{6}$$

The r_i^k determines the lower and upper limits of integration on ith dimension.

Thus, the set of parameters of our algorithms then becomes $\mathbf{r}^k$. Instead of finding r_i^k using exhaustive search, we propose to find a reasonable estimate of r_i^k using a sequence $t_i^k(l)$, $l = 1, 2, \ldots$, as follows. Given an initial value $t_i^k(0)$, we use an iterative method to find an approximation of r_i^k as follows

$$t_i^k(l) = t_i^k(l-1) - \frac{\eta(\mathbf{x}^k, t_i^k(l-1))}{\beta^k(\mathbf{x}^k, t_i^k(l-1))}, \quad l > 0, \tag{7}$$

where $\beta^k = \partial\eta(\mathbf{x}^k, \mathbf{r}^k)/\partial r_i^k$ and l is the iterative number. For an iteration number l^N, we then approximate r_i^k by $t_i^k(l^N)$ (and $d_i^k = r_i^k$).

As it can be seen, the method of finding d_i^k is based on the Newton method of finding roots of equation in one dimension. We emphasize that both functions η and β^k do not involve any evaluation of the derivative of the objective function G. The formula for the high-dimensional case can be expressed in the same fashion.

For simplicity, we now state the algorithm using the proposed method for two-dimensional function.

Step 1. For a given (x_1^k, x_2^k) and α^k, compute $t_1^k(l)$ by the iterative numerical method defined by Eq. (7) for approximating d_1^{k+1}. Set $d_1^{k+1} = t_1^k(l^N)$ for a large value of l, say l^N.

Step 2. Find the new value of x_1^{k+1} by using Eq. (5) with α^k and d_1^{k+1} obtained in Step 1.

Step 3. Update α^{k+1} by using

$$\alpha_1^{\mathrm{Cor}} = \int_{x_1^k - d_1^k}^{x_1^k + d_1^k} \int_{x_2^k - d_2^k}^{x_2^k + d_2^k} \frac{x_1 - x_1^{k+1} + \mathrm{Sign} * (x_1^{k+1} - x_1^k)}{x_1^{k+1} + \mathrm{Sign} * (x_1^{k+1} - x_1^k)} G(x_1, x_2)\, dx_1\, dx_2, \tag{8}$$

where $\mathrm{Sign} = \mathrm{sign}(x_1^{k+1} - x_1^k)$. Set $\alpha^{k+1} = \min(\alpha^k + \alpha_1^{\mathrm{Cor}}, \alpha_1^k)$ if $\alpha^k + \alpha_1^{\mathrm{Cor}} > 0$.

Step 4. Find d_2^{k+1} by using $t_2^k(l)$ in Eq. (7) with x_1^{k+1} obtained in Step 2 and α^{k+1} obtained in Step 3.

Step 5. Find the new level value of x_2^{k+1} from (5) with d_1^{k+1}, d_2^{k+1} and x_1^{k+1}.

Please cite this article as: X. Wang et al., A derivative-free optimization algorithm based on conditional moments, J. Math. Anal. Appl. (2006), doi:10.1016/j.jmaa.2006.08.091

Step 6. Update α^{k+1} by using

$$\alpha_2^{\mathrm{Cor}} = \int_{x_1^{k+1}-d_1^k}^{x_1^{k+1}+d_1^k} \int_{x_2^{k+1}-d_2^k}^{x_2^{k+1}+d_2^k} \frac{x_2 - x_2^{k+1} + \mathrm{Sign}\cdot(x_2^{k+1} - x_2^k)}{x_2^{k+1} + \mathrm{Sign}\cdot(x_2^{k+1} - x_2^k)} G(x_1, x_2)\, dx_1\, dx_2, \tag{9}$$

where $\mathrm{Sign} = \mathrm{sign}(x_2^{k+1} - x_2^k)$. Set $\alpha^{k+1} = \min(\alpha^{k+1} + \alpha_2^{\mathrm{Cor}}, \alpha^{k+1})$ if $\alpha^{k+1} + \alpha_2^{\mathrm{Cor}} > 0$.

Step 7. Repeat Steps 1–6 until convergence criterion is satisfied.

Remark. In the algorithm, the local integrations in Eqs. (8) and (9) are evaluated by numerical quadratures. For general objective functions, we can obtain accurate results by using the composite Simpson quadrature. Furthermore, the Gaussian quadratures and the composite quadratures of Gaussian types could obtain very effective and highly accurate values for high-dimensional functions (see [3,18]).

3. Theoretical properties of the method

In this section, we will establish the theoretical properties of the method. We prove that the algorithm will generate a sequence $\mathbf{x}_k$ such that $G(\mathbf{x}_k)$ is a non-decreasing sequence. We also establish the convergence and convergence rate for the proposed algorithm.

3.1. Non-decreasing property in monotone neighbourhoods

We first show that the size of $B(\mathbf{x}, d)$ is a non-decreasing function of α for any fixed position of $\mathbf{x}^k$.

Lemma 3.1. *For $\alpha_1, \alpha_2 > 0$, let $\int_{B(\mathbf{x}^k, d_1)} G(\mathbf{y})\, d\mathbf{y} = \alpha_1$ and $\int_{B(\mathbf{x}^k, d_2)} G(\mathbf{y})\, d\mathbf{y} = \alpha_2$. If $\alpha_1 > \alpha_2$, then for any given $\mathbf{x} \in R^m$, $d(\alpha_1) \geqslant d(\alpha_2)$.*

Proof. For any given α_1 and α_2 such that $\alpha_1 > \alpha_2$, let us assume that $d(\alpha_2) \geqslant d(\alpha_1)$. By (4), it follows that

$$\begin{aligned}\alpha_2 - \alpha_1 &= \int_{B(\mathbf{x}^k, d_2)} G(\mathbf{y})\, d\mathbf{y} - \int_{B(\mathbf{x}^k, d_1)} G(\mathbf{y})\, d\mathbf{y} \\ &= \int_{\{\mathbf{y}:\, d_1 \leqslant \|\mathbf{y}-\mathbf{x}^k\| \leqslant d_2\}} G(\mathbf{y})\, d\mathbf{y} \geqslant 0.\end{aligned}$$

The above inequality follows because $G(\mathbf{x}) \geqslant 0$. This is a contradiction. □

We can also show that the new estimate lies within the neighbourhood of $\mathbf{x}^k$.

Lemma 3.2. *For any given $\alpha > 0$ and $\mathbf{x}^k$, we have*

$$\mathbf{x}^{k+1} \in B\big(\mathbf{x}^k, d\big).$$

Proof. By the mean value theorem, it follows that

$$\frac{1}{\alpha}\int\limits_{B(\mathbf{x}^k,d)} \mathbf{x}G(\mathbf{x})\,d\mathbf{x} = \mathbf{x}_k^*\frac{1}{\alpha}\int\limits_{B(\mathbf{x}^k,d)} G(\mathbf{x})\,d\mathbf{x} = \mathbf{x}_k^*,$$

where $\mathbf{x}_k^* \in B(\mathbf{x}^k, d)$. □

Then, we establish the first important property of the algorithm. The following theorem shows that our algorithm will generate a strictly increasing sequence $G(\mathbf{x}^k)$, $k = 1, 2, \ldots, n$, if the derivative is monotone in a local region. This implies that algorithm will try to move up to the local optimum. Although the derivative was not used in finding the direction, the algorithm can still find the direction such that it moves to a "higher" ground.

Theorem 3.1. *For any given α and $\mathbf{x}^k$ and a continuous function $G(\mathbf{x})$, if $\frac{\partial G(\mathbf{x})}{\partial x_i} \neq 0$ on $B(\mathbf{x}^k, d)$ for $i = 1, \ldots, n$, then*

$$G\big(\mathbf{x}^{k+1}\big) > G\big(\mathbf{x}^k\big). \tag{10}$$

Proof. Observe that

$$\begin{aligned}
&\big(x_1^{k+1} - x_1^k\big)\alpha \\
&\quad= \int_{-d}^{d} \int_{-\sqrt{d^2-y_1^2}}^{\sqrt{d^2-y_1^2}} \cdots \int_{-\sqrt{d^2-y_1^2-\cdots-y_{n-1}^2}}^{\sqrt{d^2-y_1^2-\cdots-y_{n-1}^2}} \\
&\qquad\cdot y_1\big[G\big(x_1^k + y_1, x_2^k + y_2, \ldots, x_n^k + y_n\big)\big]\,dy_n\,dy_{n-1}\cdots dy_1 \\
&\quad= \int_{0}^{d} \int_{-\sqrt{d^2-y_1^2}}^{\sqrt{d^2-y_1^2}} \cdots \int_{-\sqrt{d^2-y_1^2-\cdots-y_{n-1}^2}}^{\sqrt{d^2-y_1^2-\cdots-y_{n-1}^2}} y_1\big[G\big(x_1^k + y_1, x_2^k + y_2, \ldots, x_n^k + y_n\big) \\
&\qquad- G\big(x_1^k - y_1, x_2^k + y_2, \ldots, x_n^k + y_n\big)\big]\,dy_n\,dy_{n-1}\cdots dy_1 \\
&\quad= \left(\int_{0}^{d} \int_{-\sqrt{d^2-y_1^2}}^{\sqrt{d^2-y_1^2}} \cdots \int_{-\sqrt{d^2-y_1^2-\cdots-y_{n-1}^2}}^{\sqrt{d^2-y_1^2-\cdots-y_{n-1}^2}} y_1\,dy_n\,dy_{n-1}\cdots dy_1\right) \\
&\qquad\cdot \big(G\big(x_1^k + \xi_1^{(1)}, x_2^k + \xi_2^{(1)}, \ldots, x_n^k + \xi_n^{(1)}\big) - G\big(x_1^k - \xi_1^{(1)}, x_2^k + \xi_2^{(1)}, \ldots, x_n^k + \xi_n^{(1)}\big)\big),
\end{aligned}$$

where $\xi^{(1)} = (\xi_1^{(1)}, \xi_2^{(1)}, \ldots, \xi_n^{(1)})' \in \{\mathbf{y} \in R^n\colon\ \|\mathbf{y}\| \leqslant \mathbf{d} \text{ and } y_1 \geqslant 0\}$, and $\mathbf{y} = (y_1, \ldots, y_n)'$.

Please cite this article as: X. Wang et al., A derivative-free optimization algorithm based on conditional moments, J. Math. Anal. Appl. (2006), doi:10.1016/j.jmaa.2006.08.091

Let

$$v(d) = \int_0^d \int_{-\sqrt{d^2-y_1^2}}^{\sqrt{d^2-y_1^2}} \cdots \int_{-\sqrt{d^2-y_1^2-\cdots-y_{n-1}^2}}^{\sqrt{d^2-y_1^2-\cdots-y_{n-1}^2}} y_1 \, dy_n \cdots dy_1.$$

It is clear that $v(d)$ is positive.

It then follows that there exists $\xi^{(i)} = (\xi_1^{(i)}, \xi_2^{(i)}, \ldots, \xi_n^{(i)})' \in \{\mathbf{y} \in R^n\colon \|\mathbf{y}\| \leqslant d \text{ and } y_i \geqslant 0\}$, $i = 2, 3, \ldots, n$, such that

$$\begin{aligned}
&\big(x_i^{k+1} - x_i^k\big) \\
&\quad = \frac{v(d)}{\alpha} \times \big[G\big(x_1^k + \xi_1^{(i)}, \ldots, x_{i-1}^k + \xi_{i-1}^{(i)}, x_i^k + \xi_i^{(i)}, x_{i+1}^k + \xi_{i+1}^{(i)}, \ldots, x_n^k + \xi_n^{(i)}\big) \\
&\qquad - G\big(x_1^k + \xi_1^{(i)}, \ldots, x_{i-1}^k + \xi_{i-1}^{(i)}, x_i^k - \xi_i^{(i)}, x_{i+1}^k + \xi_{i+1}^{(i)}, \ldots, x_n^k + \xi_n^{(i)}\big)\big].
\end{aligned}$$

Since the objective function is monotone on $B(\mathbf{x}^k, d^k)$, therefore the sign of $x_i^{k+1} - x_i^k$ is determined by whether the objective function is increasing or decreasing. Therefore,

$$\frac{\partial G}{\partial x_i}\big(\mathbf{x}^k + \theta\big(\mathbf{x}^{k+1} - \mathbf{x}^k\big)\big)\big(\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\big) > 0, \tag{11}$$

where $\theta \in [0, 1]$, $i = 1, \ldots, n$.

By the Taylor expansion, we know that:

$$G\big(\mathbf{x}^{k+1}\big) = G\big(\mathbf{x}^k\big) + \sum_{i=1}^n \frac{\partial G}{\partial x_i}\big(\mathbf{x}^k + \theta\big(\mathbf{x}^{k+1} - \mathbf{x}^k\big)\big)\big(\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\big), \quad 0 \leqslant \theta \leqslant 1. \tag{12}$$

It follows that $G(\mathbf{x}^{k+1}) > G(\mathbf{x}^k)$. □

If the derivative is close to zero, the algorithm can still produce a strictly increasing sequence $G(\mathbf{x}^k)$, $i = 1, 2, \ldots, n$, for some given α. The next theorem formally establishes the result.

Theorem 3.2. *Assume the following*:

(i) *The optimum is contained in a bounded and closed set O. For any given $\varepsilon > 0$, $D(\varepsilon) \triangleq O \cap \bigcup_{i=1}^n \{\mathbf{x} \in R^n\colon |\frac{\partial G}{\partial x_i}(\mathbf{x})| \geqslant \varepsilon\}$ is closed*;

(ii) *The partial derivatives $\frac{\partial G(\mathbf{x})}{\partial x_i}$ is continuous with respect to $\mathbf{x}$ in R^n, $i = 1, 2, \ldots, n$.*

Then, there exists $\bar{\alpha} > 0$, such that for any $\alpha \in (0, \bar{\alpha}]$, and $\mathbf{x}^k \in D(\varepsilon)$,

$$G\big(\mathbf{x}^{k+1}(\alpha)\big) > G\big(\mathbf{x}^k(\alpha)\big).$$

Proof. For any given $\varepsilon > 0$, $D(\varepsilon)$ is bounded and closed. It follows that $\frac{\partial G(\mathbf{x})}{\partial x_i}$ is uniformly continuous in $D(\varepsilon)$ with respect to x, i.e. $\exists \delta(\varepsilon) > 0$ such that, $\forall \mathbf{x} \in D(\varepsilon)$, if $\|\mathbf{x}' - \mathbf{x}\| < \delta$, then $|\frac{\partial G(\mathbf{x}')}{\partial x_i} - \frac{\partial G(\mathbf{x})}{\partial x_i}| < \frac{\varepsilon}{2n}$. By assumption (ii), $G(\mathbf{x})$ is continuous with respect to $\mathbf{x}$ in R^n. Then, $\forall \mathbf{x} \in D(\varepsilon)$, we define

$$\alpha(\mathbf{x}) = \int_{\|\mathbf{y}-\mathbf{x}\| \leqslant \delta/2} G(\mathbf{y}) \, d\mathbf{y}. \tag{13}$$

Please cite this article as: X. Wang et al., A derivative-free optimization algorithm based on conditional moments, J. Math. Anal. Appl. (2006), doi:10.1016/j.jmaa.2006.08.091

It follows that $\alpha(\mathbf{x})$ is continuous with respect to $\mathbf{x}$. We note that $\alpha(\mathbf{x}) > 0$. Otherwise, assume that $\alpha(\mathbf{x}) = 0$, i.e. $\int_{\|\mathbf{y}-\mathbf{x}\|\leqslant\delta/2} G(\mathbf{y})\,d\mathbf{y} = 0$. Since $G(\mathbf{y}) \geqslant 0$, then $G(\mathbf{y}) = 0$ for any $\mathbf{y}$ such that $\mathbf{y} \in \{\mathbf{y} \in R^n\colon\ \|\mathbf{x}-\mathbf{y}\| \leqslant \frac{\delta}{2}\}$. Thus, $\frac{\partial G}{\partial x_i}(\mathbf{x}) = 0$, $i = 1, 2, \ldots, n$, i.e. $\mathbf{x} \in R^n - D(\varepsilon)$. This is contradiction.

Since $D(\varepsilon)$ is bounded and closed, therefore $\alpha(\mathbf{x})$ has a minimum value in the field $D(\varepsilon)$, say $\tilde{\alpha}$. Let $\int_{\|\mathbf{x}-\mathbf{y}\|\leqslant\bar{d}} G(\mathbf{y})\,d\mathbf{y} = \tilde{\alpha}$, $\mathbf{x} \in D(\varepsilon)$, then $\bar{d}(\mathbf{x}) \leqslant \delta/2$, $\mathbf{x} \in D(\varepsilon)$.

Note that

$$
\begin{aligned}
&x_1^{k+1} - x_1^k \\
&\quad= \frac{1}{\alpha} \int_{-\bar{d}(x^k)}^{\bar{d}(x^k)} \int_{-\sqrt{\bar{d}(x^k)^2-y_1^2}}^{\sqrt{\bar{d}(x^k)^2-y_1^2}} \cdots \int_{-\sqrt{\bar{d}(x^k)^2-y_1^2-\cdots-y_{n-1}^2}}^{\sqrt{\bar{d}(x^k)^2-y_1^2-\cdots-y_{n-1}^2}} \\
&\qquad\cdot y_1 G\bigl(x_1^k + y_1, x_2^k + y_2, \ldots, x_n^k + y_n\bigr)\,dy_n\,dy_{n-1}\cdots dy_1 \\
&\quad= \frac{1}{\alpha} \int_{-\bar{d}(x^k)}^{\bar{d}(x^k)} \int_{-\sqrt{\bar{d}(x^k)^2-y_1^2}}^{\sqrt{\bar{d}(x^k)^2-y_1^2}} \cdots \int_{-\sqrt{\bar{d}(x^k)^2-y_1^2-\cdots-y_{n-1}^2}}^{\sqrt{\bar{d}(x^k)^2-y_1^2-\cdots-y_{n-1}^2}} y_1\bigl[G\bigl(x_1^k, x_2^k + y_2, \ldots, x_n^k + y_n\bigr) \\
&\qquad+ y_1 \frac{\partial G}{\partial x_1}\bigl(x_1^k + \theta y_1, x_2^k + y_2, \ldots, x_n^k + y_n\bigr)\bigr]\,dy_n\,dy_{n-1}\cdots dy_1 \\
&\quad= \frac{1}{\alpha} \int_{-\bar{d}(x^k)}^{\bar{d}(x^k)} y_1 \int_{-\sqrt{\bar{d}(x^k)^2-y_1^2}}^{\sqrt{\bar{d}^2-y_1^2}} \cdots \int_{-\sqrt{\bar{d}(x^k)^2-y_1^2-\cdots-y_{n-1}^2}}^{\sqrt{\bar{d}(x^k)^2-y_1^2-\cdots-y_{n-1}^2}} \\
&\qquad\cdot G\bigl(x_1^k, x_2^k + y_2, \ldots, x_n^k + y_n\bigr)\,dy_n\,dy_{n-1}\cdots dy_1 \\
&\qquad+ \frac{1}{\alpha} \int_{-\bar{d}(x^k)}^{\bar{d}(x^k)} \int_{-\sqrt{\bar{d}(x^k)^2-y_1^2}}^{\sqrt{\bar{d}(x^k)^2-y_1^2}} \cdots \int_{-\sqrt{\bar{d}(x^k)^2-y_1^2-\cdots-y_{n-1}^2}}^{\sqrt{\bar{d}(x^k)^2-y_1^2-\cdots-y_{n-1}^2}} \\
&\qquad\cdot y_1^2 \frac{\partial G}{\partial x_1}\bigl(x_1^k + \theta(y_1)y_1, x_2^k + y_2, \ldots, x_n^k + y_n\bigr)\,dy_n\cdots dy_1,
\end{aligned}
$$

where $\theta(y_1) \in [0, 1]$.

Let

$$
\begin{aligned}
h(y_1) = y_1 &\int_{-\sqrt{\bar{d}(x^k)^2-y_1^2}}^{\sqrt{\bar{d}(x^k)^2-y_1^2}} \cdots \int_{-\sqrt{\bar{d}(x^k)^2-y_1^2-\cdots-y_{n-1}^2}}^{\sqrt{\bar{d}(x^k)^2-y_1^2-\cdots-y_{n-1}^2}} \\
&\cdot G\bigl(x_1^k, x_2^k + y_2, \ldots, x_n^k + y_n\bigr)\,dy_n\,dy_{n-1}\cdots dy_2.
\end{aligned}
$$

Since h is odd function, thus $\int_{-\bar{d}(x^k)}^{\bar{d}(x^k)} h(y_1)\,dy_1 = 0$. It then follows that

Please cite this article as: X. Wang et al., A derivative-free optimization algorithm based on conditional moments, J. Math. Anal. Appl. (2006), doi:10.1016/j.jmaa.2006.08.091

$$x_1^{k+1} - x_1^k = \frac{1}{\alpha} \int_{-\bar{d}(x^k)}^{\bar{d}(x^k)} \int_{-\sqrt{\bar{d}(x^k)^2 - y_1^2}}^{\sqrt{\bar{d}(x^k)^2 - y_1^2}} \cdots \int_{-\sqrt{\bar{d}(x^k)^2 - y_1^2 - \cdots - y_{n-1}^2}}^{\sqrt{\bar{d}(x^k)^2 - y_1^2 - \cdots - y_{n-1}^2}} \cdot y_1^2 \frac{\partial G}{\partial x_1}\left(x_1^k + \theta(y_1) y_1, x_2^k + y_2, \ldots, x_n^k + y_n\right) dy_n \, dy_{n-1} \cdots dy_1,$$

where $\theta(y_1) \in [0, 1]$.

In general, we have

$$x_i^{k+1} - x_i^k = \frac{1}{\alpha} \int_{-\bar{d}(x^k)}^{\bar{d}(x^k)} \int_{-\sqrt{\bar{d}(x^k)^2 - y_1^2}}^{\sqrt{\bar{d}(x^k)^2 - y_1^2}} \cdots \int_{-\sqrt{\bar{d}(x^k)^2 - y_1^2 - \cdots - y_{n-1}^2}}^{\sqrt{\bar{d}(x^k)^2 - y_1^2 - \cdots - y_{n-1}^2}} \cdot y_i^2 \cdot \frac{\partial G}{\partial x_i}\left(x_1^k + y_1, x_2^k + y_2, \ldots, x_{i-1}^k + y_{i-1}^k, x_i^k + \theta(y_i) y_i, x_{i+1}^k + y_{i+1}^k, \ldots, x_n^k + y_n\right) \cdot dy_n \, dy_{n-1} \cdots dy_1,$$

where $\theta(y_i) \in [0, 1]$.

For any i, we define

$$C\left(\bar{d}\left(\mathbf{x}^k\right)\right) = \int_{-\bar{d}(\mathbf{x}^k)}^{\bar{d}(\mathbf{x}^k)} \int_{-\sqrt{\bar{d}(\mathbf{x}^k)^2 - y_1^2}}^{\sqrt{\bar{d}(\mathbf{x}^k)^2 - y_1^2}} \cdots \int_{-\sqrt{\bar{d}(\mathbf{x}^k)^2 - y_1^2 - \cdots - y_{n-1}^2}}^{\sqrt{\bar{d}(\mathbf{x}^k)^2 - y_1^2 - \cdots - y_{n-1}^2}} y_i^2 \, dy_n \cdots dy_1. \tag{14}$$

(a) First, consider the case in which it satisfies

$$\bigcup_i \left(B\left(\mathbf{x}^k, \bar{d}\right) \cap \left\{ \mathbf{x} \in R \colon \frac{\partial G}{\partial x_i}\left(\mathbf{x}^k\right) = 0 \right\} \right) \neq \phi. \tag{15}$$

Without loss of generality, we assume that

$$B\left(\mathbf{x}^k, \bar{d}\right) \cap \left\{ \mathbf{x} \in R : \frac{\partial G}{\partial x_1}\left(\mathbf{x}^k\right) = 0 \right\} \neq \phi.$$

It suffices to only consider the following case in which

$$\frac{\partial G}{\partial x_1}\left(\mathbf{x}^k\right) \geqslant \varepsilon. \tag{16}$$

Since when $\mathbf{y} \in B(\mathbf{x}^k, \bar{d})$, $|\frac{\partial G}{\partial x_1}(\mathbf{y}) - \frac{\partial G}{\partial x_1}(\mathbf{x}^k)| < \frac{1}{2n}\varepsilon$, therefore

$$\frac{\partial G}{\partial x_1}(\mathbf{y}) > \frac{2n-1}{2n}\varepsilon. \tag{17}$$

We then have

$$\mathbf{x}_1^{k+1} - \mathbf{x}_1^k > \frac{1}{\alpha} \cdot \frac{2n-1}{2n} \varepsilon \int_{-\bar{d}(\mathbf{x}^k)}^{\bar{d}(\mathbf{x}^k)} \int_{-\sqrt{\bar{d}(\mathbf{x}^k)^2 - y_1^2}}^{\sqrt{\bar{d}(\mathbf{x}^k)^2 - y_1^2}} \cdots \int_{-\sqrt{\bar{d}(\mathbf{x}^k)^2 - y_1^2 - \cdots - y_{n-1}^2}}^{\sqrt{\bar{d}(\mathbf{x}^k)^2 - y_1^2 - \cdots - y_{n-1}^2}} y_1^2 \, dy_n \, dy_{n-1} \cdots dy_1.$$

Please cite this article as: X. Wang et al., A derivative-free optimization algorithm based on conditional moments, J. Math. Anal. Appl. (2006), doi:10.1016/j.jmaa.2006.08.091

It then follows that

$$\mathbf{x}_1^{k+1} - \mathbf{x}_1^k > \frac{1}{\alpha} \cdot \frac{2n-1}{2n} \cdot \varepsilon \cdot C\big(\bar{d}\big(\mathbf{x}^k\big)\big). \tag{18}$$

By (15), there exists $x^{(p)} \in B(\bar{\alpha}, x^k)$, such that $\frac{\partial G}{\partial x_p}(x^{(p)}) = 0$.

Therefore, if $y \in B(\bar{\alpha}, x^k)$, it then follows that

$$\begin{aligned}\left|\frac{\partial G}{\partial \mathbf{x}_p}(\mathbf{y})\right| &= \left|\frac{\partial G}{\partial \mathbf{x}_p}(\mathbf{y}) - \frac{\partial G}{\partial \mathbf{x}_p}\big(\mathbf{x}^k\big) + \frac{\partial G}{\partial \mathbf{x}_p}\big(\mathbf{x}^k\big) - \frac{\partial G}{\partial \mathbf{x}_p}\big(\mathbf{x}^{(p)}\big)\right| \\ &\leqslant \left|\frac{\partial G}{\partial x_p}(\mathbf{y}) - \frac{\partial G}{\partial x_i}\big(\mathbf{x}^k\big)\right| + \left|\frac{\partial G}{\partial x_p}\big(\mathbf{x}^k\big) - \frac{\partial G}{\partial x_p}\big(\mathbf{x}^{(p)}\big)\right| < \frac{\varepsilon}{2n} + \frac{\varepsilon}{2n} = \frac{\varepsilon}{n}.\end{aligned}$$

Since $|\frac{\partial G}{\partial x_p}(\mathbf{y})| < \frac{1}{n}\varepsilon$ for $\mathbf{y} \in B(\mathbf{x}^k, \bar{d})$, therefore,

$$\begin{aligned}\big|\mathbf{x}_p^{k+1} - \mathbf{x}_p^k\big| &\leqslant \frac{1}{\alpha} \int_{-\bar{d}(\mathbf{x}^k)}^{\bar{d}(\mathbf{x}^k)} \int_{-\sqrt{\bar{d}(\mathbf{x}^k)^2 - \mathbf{y}_1^2}}^{\sqrt{\bar{d}(\mathbf{x}^k)^2 - \mathbf{y}_1^2}} \cdots \int_{-\sqrt{\bar{d}(\mathbf{x}^k)^2 - y_1^2 - \cdots - y_{n-1}^2}}^{\sqrt{\bar{d}(\mathbf{x}^k)^2 - y_1^2 - \cdots - y_{n-1}^2}} y_p^2 \\ &\quad \cdot \left|\frac{\partial G}{\partial x_p}\big(x_1^k + y_1, x_2^k + y_2, \ldots, x_{p-1}^k + y_{p-1}^k, x_p^k \right. \\ &\quad \left. + \theta(y_p) y_p, x_{p+1}^k + y_{p+1}^k, \ldots, x_n^k + y_n\big)\right| dy_n\, dy_{n-1} \cdots dy_1 \\ &< \frac{1}{\alpha}\frac{1}{n}\varepsilon \int_{-\bar{d}(x^k)}^{\bar{d}(x^k)} \int_{-\sqrt{\bar{d}(x^k)^2 - y_1^2}}^{\sqrt{\bar{d}(x^k)^2 - y_1^2}} \cdots \int_{-\sqrt{\bar{d}(x^k)^2 - y_1^2 - \cdots - y_{n-1}^2}}^{\sqrt{\bar{d}(x^k)^2 - y_1^2 - \cdots - y_{n-1}^2}} y_p^2\, dy_n\, dy_{n-1} \cdots dy_1 \\ &= \frac{1}{\alpha}\frac{1}{n}\varepsilon C\big(\bar{d}\big(\mathbf{x}^k\big)\big).\end{aligned}$$

Let $\mathcal{E} = \{p \mid \frac{\partial G}{\partial \mathbf{x}_p}(\mathbf{x}^k) = 0\}$. By Taylor expansion and (11),

$$\begin{aligned}&G\big(\mathbf{x}^{k+1}\big) - G\big(\mathbf{x}^k\big) \\ &\quad = \frac{\partial G}{\partial x_1}\big(x^k + \theta\big(\mathbf{x}^{k+1} - \mathbf{x}^k\big)\big)\big(\mathbf{x}_1^{k+1} - \mathbf{x}_1^k\big) + \sum_{i=2}^{n} \frac{\partial G}{\partial x_i}\big(\mathbf{x}^k + \theta\big(\mathbf{x}^{k+1} - \mathbf{x}^k\big)\big)\big(\mathbf{x}_i^{k+1} - \mathbf{x}_i^k\big) \\ &\quad > \frac{1}{\alpha} \cdot \frac{2n-1}{2n}\varepsilon \cdot \frac{2n-1}{2n}\varepsilon \cdot C\big(\bar{d}\big(\mathbf{x}^k\big)\big) - \sum_{p \in \mathcal{E}} \left|\frac{\partial G}{\partial x_p}\big(x^k + \theta\big(x^{k+1} - x^k\big)\big)\right| \cdot \big|x_p^{k+1} - x_p^k\big| \\ &\quad > \frac{1}{\alpha} \cdot \left(\frac{2n-1}{2n}\right)^2 \cdot \varepsilon^2 C\big(\bar{d}\big(\mathbf{x}^k\big)\big) - \frac{1}{\alpha} \cdot \frac{1}{n}\varepsilon \cdot C\big(\bar{d}\big(\mathbf{x}^k\big)\big) \cdot \frac{1}{n}\varepsilon \cdot (n-1) \\ &\quad = \frac{1}{\alpha} \cdot \varepsilon^2 \cdot C\big(\bar{d}\big(\mathbf{x}^k\big)\big) \cdot \left(\frac{4n^2 - 8n + 5}{4n^2}\right).\end{aligned}$$

Please cite this article as: X. Wang et al., A derivative-free optimization algorithm based on conditional moments, J. Math. Anal. Appl. (2006), doi:10.1016/j.jmaa.2006.08.091

(b) Second, if $B(\mathbf{x}^k, \bar{d}) \cup \bigcap_{i=1}^n \{x \in R^n\colon \frac{\partial G}{\partial x_i}(\mathbf{x}^k) = 0\} = \phi$, since $\frac{\partial G}{\partial x_i}$ is continuous, therefore, $\frac{\partial G}{\partial x_i}(\mathbf{y}) > 0$ (or < 0), where $y \in B(\mathbf{x}^k, \bar{d})$. By (11) and the Taylor expansion of $G(\mathbf{x}^{k+1})$, we then have

$$G\big(\mathbf{x}^{k+1}\big) - G\big(\mathbf{x}^k\big) = \sum_{i=1}^{n} \frac{\partial G}{\partial x_i}\big(\mathbf{x}^k + \theta\big(\mathbf{x}^{k+1} - \mathbf{x}^k\big)\big)\big(x_i^{k+1} - x_i^k\big)$$
$$> \frac{1}{\alpha} \cdot \varepsilon^2 \cdot C\big(\bar{d}\big(\mathbf{x}^k\big)\big) \cdot \left(\frac{4n^2 - 8n + 5}{4n^2}\right).$$

Since $C(\bar{d}(\mathbf{x}^k))$ is positive and $4n^2 - 8n + 5 > 0$ when $n \geqslant 1$. It then follows that

$$G\big(\mathbf{x}^{k+1}\big) - G\big(\mathbf{x}^k\big) > 0. \qquad \square$$

The above theorem only characterizes the behaviour of the algorithm when the derivative is not zero or its absolute value is bounded from below. However, it does not provide any assurance that the algorithm will enter into a small neighbourhood of the optima. The following theorem in the next subsection shows that at least some of the estimates to the optima generated by the algorithm will have the first-order derivative that is close to zero.

3.2. Convergence of the method

In this part we will discuss the convergence theorems for our method. Basically, we prove that there exists a path of convergence to the true global optimum.

Theorem 3.3. *Under the assumptions of Theorem* 3.2*, there exists a sub-sequence* $\{\mathbf{x}^{m_k}(\alpha)\}$*, where* $\alpha \in (0, \bar{\alpha}]$*, which satisfies*: *Any* $\varepsilon > 0$*,* $\exists K(\varepsilon) > 0$*, when* $k > K(\varepsilon)$*,* $\mathbf{x}^{m_k} \in D(\varepsilon)^c$*, where* $D(\varepsilon)^c = R^n - D(\varepsilon)$.

Proof. Assume that no such sub-sequence exists, then there must exist $M > 0$ such that when $m > M$, we have $\mathbf{x}^m \in D(\varepsilon)$. Denote $G(\mathbf{x}^m)$ by $\mathbf{y}_m$. By Theorem 3.4, when $m > M$, $\{\mathbf{y}_m\}$ is increasing.

Note that G is continuous since it is differentiable. Since $D(\varepsilon)$ is closed and bounded, thus G is bounded in $D(\varepsilon)$. It then follows that the sequence $\{\mathbf{y}_m\}$ is convergent. Denote the limit by $\tilde{\mathbf{y}}$, we must have $y_k \leqslant \tilde{\mathbf{y}}$ when $k > M$. Since $D(\varepsilon)$ is bounded, therefore $\{\mathbf{x}^m\}$ has a convergent sub-sequence. Assume it is $\{\mathbf{x}^{k_l}\}$. Denote the limit by $\tilde{\mathbf{x}}$. $D(\varepsilon)$ is closed, so $\tilde{\mathbf{x}} \in D(\varepsilon)$.

Since G is continuous, we have

$$G(\tilde{x}) = G\Big(\lim_{l\to\infty} \mathbf{x}^{k_l}\Big) = \lim_{l\to\infty} G\big(\mathbf{x}^{k_l}\big) = \lim_{l\to\infty} \mathbf{y}^{k_l} = \tilde{\mathbf{y}}.$$

By Theorem 3.4, $G(T(\tilde{\mathbf{x}}, d(\bar{\alpha}))) > G(\tilde{\mathbf{x}})$, where T is defined in (5).

Observe that $G \circ T$ is continuous with respect to $\mathbf{x}$ since G and T are continuous. Let $\varepsilon = G(T(\tilde{\mathbf{x}}, d(\bar{\alpha}))) - G(\tilde{\mathbf{x}})$. Thus, $\exists L > 0$, s.t. $l > L$, $|G(T(\tilde{\mathbf{x}}, \bar{\alpha})) - G(T(\mathbf{x}^{m_l}, d(\bar{\alpha})))| < \varepsilon$, i.e. $G(\mathbf{x}^{m_l+1}) = G(T(\mathbf{x}^{m_l}, d(\bar{\alpha}))) > G(T(\tilde{\mathbf{x}}, d(\bar{\alpha}))) - \varepsilon = G(\tilde{\mathbf{x}})$.
Therefore, $\mathbf{y}_{m_l+1} > \tilde{\mathbf{y}}$ when $l > L$. It is a contradiction. $\square$

We then can establish the convergence rate of the proposed algorithm.

Please cite this article as: X. Wang et al., A derivative-free optimization algorithm based on conditional moments, J. Math. Anal. Appl. (2006), doi:10.1016/j.jmaa.2006.08.091

Theorem 3.4. *Under the assumptions of Theorem* 3.2*, if* $\mathbf{x}^k \in D(\varepsilon)$ *where* $D(\varepsilon)$ *is defined as in Theorem* 3.2*, then there exists a constant* $r \in (0,1)$ *such that*

$$G(\mathbf{x}^*) - G\big(\mathbf{x}^{k+1}\big) < r \cdot \big(G(\mathbf{x}^*) - G\big(\mathbf{x}^k\big)\big), \tag{19}$$

where G achieves its maximum value at the point $\mathbf{x}^*$.

Proof. From the proof of Theorem 3.2, we know that:

$$G\big(\mathbf{x}^{k+1}\big) - G\big(\mathbf{x}^k\big) > \frac{1}{\alpha} \cdot \varepsilon^2 \cdot C\big(d\big(\mathbf{x}^k\big)\big) \cdot \left(\frac{4n^2 - 8n + 5}{4n^2}\right).$$

When α fixed, then by the Theorem of Implicit Function, it follows that d is continuous with respect to x, $\mathbf{x} \in D(\varepsilon)$. Since $D(\varepsilon)$ is closed and bounded, therefore d has minimum value when $\mathbf{x} \in D(\varepsilon)$, denoted by d_0. Let $C_0 = \frac{1}{\alpha} \cdot \varepsilon^2 \cdot C(d_0) \cdot (\frac{4n^2-8n+5}{4n^2})$. Thus, $G(\mathbf{x}^{k+1}) - G(\mathbf{x}^k) > C_0$. Note that $G(\mathbf{x}^{k+1}) \leqslant G(\mathbf{x}^*)$, so $0 < C_0 < G(\mathbf{x}^*) - G(\mathbf{x}^k)$. Thus there exists $c \in (0,1)$, s.t. $C_0 = c \cdot (G(\mathbf{x}^*) - G(\mathbf{x}^k))$. Therefore,

$$0 \leqslant G(\mathbf{x}^*) - G\big(\mathbf{x}^{k+1}\big) < G(\mathbf{x}^*) - G\big(\mathbf{x}^k\big) - C_0 = r \cdot \big(G(\mathbf{x}^*) - G\big(\mathbf{x}^k\big)\big),$$

where $r = 1 - c$. □

Lemma 3.3. *Let* $\mathbf{x}^*$ *be the maximizer of G and* $\nabla G(\mathbf{x}^*) = 0$*. If the function* $G(\cdot)$ *is twice continuously differentiable and there exist constants* $0 < m \leqslant M < \infty$ *such that for all* $\mathbf{x}, \mathbf{y} \in R^n$*,*

$$-M\|\mathbf{y}\|^2 \leqslant \big\langle \mathbf{y}, G_{\mathbf{xx}}(\mathbf{x})\mathbf{y}\big\rangle \leqslant -m\|y\|^2, \tag{20}$$

then, we must have

$$\frac{m}{2}\|\mathbf{x}^* - \mathbf{x}\|^2 \leqslant G(\mathbf{x}^*) - G(\mathbf{x}) \leqslant \frac{M}{2}\|\mathbf{x}^* - \mathbf{x}\|^2, \tag{21}$$

where $\mathbf{x}^*$ *is the maximizer of* $G(\mathbf{x})$.

Proof. Note that

$$G(\mathbf{x}) - G(\mathbf{x}^*) = \big\langle \nabla G(\mathbf{x}^*), (\mathbf{x} - \mathbf{x}^*)\big\rangle + \frac{1}{2}\big\langle \mathbf{x} - \mathbf{x}^*, G_{\mathbf{xx}}\big(\mathbf{x} + \theta(\mathbf{x} - \mathbf{x}^*)\big)(\mathbf{x} - \mathbf{x}^*)\big\rangle$$

for some $\theta \in [0,1]$. □

Finally, we can prove that the algorithm converges *R-linearly* before entering $D^c(\varepsilon)$. Detailed discussions on different rate of convergence for algorithms can be found in Polak [14].

Theorem 3.5. *Suppose the assumptions in Theorem* 3.2 *and Lemma* 3.3 *hold. Let* $\bar{\varepsilon}$ *and* $D(\varepsilon)$ *be the same as defined in Theorem* 3.2*. Let* $\{\mathbf{x}^k(\alpha)\}$ *be a sequence defined by our algorithm, where* $\alpha \in (0, \bar{\alpha}]$*. If* $\mathbf{x}^k \in D(\varepsilon)$ *for* $k \leqslant K$*, then there exists* $c \in (0,1)$*, s.t.* $\|\mathbf{x}^* - \mathbf{x}^k\| < c^k[\frac{2}{m}(G(\mathbf{x}^*) - G(\mathbf{x}^0))]^{1/2}$*, where* $\mathbf{x}^*$ *is the maximizer of the function* $G(\mathbf{x})$.

Proof. From Theorem 3.4, we know that $\exists r \in (0,1)$, s.t. $G(\mathbf{x}^*) - G(\mathbf{x}^{k+1}) < r \cdot (G(\mathbf{x}^*) - G(\mathbf{x}^k))$. By Lemma 3.3, it follows that $\frac{m}{2}\|\mathbf{x}^* - \mathbf{x}^{k+1}\|^2 \leqslant G(\mathbf{x}^*) - G(\mathbf{x}^{k+1}) < r \cdot (G(\mathbf{x}^*) - G(\mathbf{x}^k))$. It follows by recursion that, for $k \leqslant K$,

$$\frac{m}{2}\big\|\mathbf{x}^* - \mathbf{x}^{k+1}\big\|^2 < r^{k+1} \cdot \big(G(\mathbf{x}^*) - G\big(\mathbf{x}^0\big)\big). \qquad \square$$

Please cite this article as: X. Wang et al., A derivative-free optimization algorithm based on conditional moments, J. Math. Anal. Appl. (2006), doi:10.1016/j.jmaa.2006.08.091

3.3. Results for unimodal functions

In this subsection, we now consider functions that are unimodal. Let x^* be the maximum for a unimodal function and define

$$E(\varepsilon) = \left\{ \mathbf{x} \in R^n \colon \left| \frac{\partial G}{\partial x_i}(\mathbf{x}) \right| \leqslant \frac{2n+1}{2n}\varepsilon,\ i = 1, \ldots, n \right\}. \tag{22}$$

We also define a family of sets: $N(\mathbf{x}^*, \varepsilon) = \{B \subset \bar{E}(\varepsilon)\colon \mathbf{x}^* \in B \text{ and } B \text{ is connected}\}$. Obviously, $\bigcup N(\mathbf{x}^*, \varepsilon)$ is connected and closed, denoted by $U(\mathbf{x}^*, \varepsilon)$. Therefore, $\bar{E}(\varepsilon) = U(\mathbf{x}^*, \varepsilon) \cup (\bar{E}(\varepsilon) - U(\mathbf{x}^*, \varepsilon))$ and $U(\mathbf{x}^*, \varepsilon)$ is the maximum connected sub-set of $\bar{E}(\varepsilon)$ which includes the point $\mathbf{x}^*$. Define

$$M_{E/U}(\varepsilon) = \sup_{\mathbf{x} \in E(\frac{2n}{2n+1}\varepsilon) - U(\mathbf{x}^*, \frac{2n}{2n+1}\varepsilon)} G(\mathbf{x}),$$

$$m_U(\varepsilon) = \inf_{x \in U(\mathbf{x}^*, \varepsilon)} G(\mathbf{x}).$$

Furthermore, we define

$$A(\mathbf{x}^*, \varepsilon) = \{t\colon m_U(\varepsilon) < t < G(\mathbf{x}^*)\}, \tag{23}$$

$$B(\mathbf{x}^*, \varepsilon) = \{t\colon M_{E/U}(\varepsilon) < t < G(\mathbf{x}^*)\}. \tag{24}$$

Theorem 3.6. *Under the assumptions of Theorem* 3.2*, if the function G is unimodal and $\exists \varepsilon^* > 0$, s.t. $G(\mathbf{x}) > 0$ where $x \in U(\mathbf{x}^*, \frac{2n}{2n+1}\varepsilon^*)$, then given $0 < \varepsilon < \varepsilon^*$ and any initial point $\mathbf{x}^0 \in G^{-1}(B(\mathbf{x}^*, \varepsilon))$, there exists $\bar{\alpha}$, $\forall \alpha \in (0, \bar{\alpha}]$, such that there exists $K(\varepsilon, \mathbf{x}^0) > 0$, when $k > K(\varepsilon, \mathbf{x}^0)$, $G(\mathbf{x}^k) \geqslant m_U(\varepsilon)$.*

Proof. Since G is unimodal, we then have $\exists \varepsilon_0 > 0$, $U(\mathbf{x}^*, \frac{2n}{2n+1}\varepsilon)$ is bounded when $\varepsilon < \varepsilon_0$.

Let $\bar{\varepsilon} = \min\{\varepsilon_0, \varepsilon^*\}$. Since $U(\mathbf{x}^*, \frac{2n}{2n+1}\bar{\varepsilon})$ is closed and bounded, therefore $D(\bar{\varepsilon}) \cup U(\mathbf{x}^*, \frac{2n}{2n+1}\bar{\varepsilon})$ is bounded and closed where $\bar{d}$ and $D(\bar{\varepsilon})$ are defined in Theorem 3.2.

We can find $\bar{\alpha} > 0$ and $\bar{d}$ as described in Theorem 3.2. It follows that, for any $\alpha \in (0, \bar{\alpha}]$, and for all $x \in D(\bar{\varepsilon}) \cup U(\mathbf{x}^*, \frac{2n}{2n+1}\bar{\varepsilon})$, if $\|\mathbf{x}' - \mathbf{x}\| < \bar{d}$, then

$$\left| \frac{\partial G(\mathbf{x}')}{\partial x_i} - \frac{\partial G(\mathbf{x})}{\partial x_i} \right| < \frac{\bar{\varepsilon}}{2n} \tag{25}$$

for $i = 1, 2, \ldots, n$.

Observe that $\mathbf{x}^0 \in U(\mathbf{x}^*, \frac{2n}{2n+1}\bar{\varepsilon}) \cup D(\bar{\varepsilon})$ since $x^0 \in G^{-1}(B(\mathbf{x}^*, \bar{\varepsilon}))$.

(1) We first consider $\mathbf{x}^0 \in D(\bar{\varepsilon})$. By Theorem 3.2, it follows that $\{\mathbf{x}^k\}$ is an increasing sequence on $D(\bar{\varepsilon})$. This implies that $G(\mathbf{x}^k) > m_U(\bar{\varepsilon})$ before the sequence enters $\bar{E}(\bar{\varepsilon})$. Therefore, by Theorem 3.5, $\exists K > 0$, s.t. $\mathbf{x}^K \in U(\mathbf{x}^*, \frac{2n}{2n+1}\varepsilon)$.

(2) Next we consider x^{K+1}.
   (i) If $\mathbf{x}^{K+1} \in U(\mathbf{x}^*, \frac{2n}{2n+1}\bar{\varepsilon})$, we then have $G(\mathbf{x}^{K+1}) \geqslant F(\varepsilon)$.
   (ii) If $\mathbf{x}^{K+1} \in D(\bar{\varepsilon})$, therefore $|\frac{\partial G}{\partial x_i}(\mathbf{x}^{K+1})| < \frac{2n+1}{2n}\bar{\varepsilon}$ since $|\frac{\partial G}{\partial x_i}(\mathbf{x}^K) - \frac{\partial G}{\partial x_i}(\mathbf{x}^{K+1})| < \frac{\bar{\varepsilon}}{2n}$ and $|\frac{\partial G}{\partial x_i}(\mathbf{x}^K)| < \bar{\varepsilon}$. Then it follows that $G(\mathbf{x}^{K+1}) \geqslant m_U(\bar{\varepsilon})$. Note that $G(\mathbf{x}^{k+1}) > G(\mathbf{x}^k)$ if $\mathbf{x}^k \in D(\bar{\varepsilon})$.

Then it follows that, for any $\epsilon < \bar{\varepsilon}$, and $k > K$, we have $G(\mathbf{x}^k) \geqslant F(\epsilon)$. □

Please cite this article as: X. Wang et al., A derivative-free optimization algorithm based on conditional moments, J. Math. Anal. Appl. (2006), doi:10.1016/j.jmaa.2006.08.091

For any sequence $\{\alpha_k\}$, we can define a new sequence $\{\mathbf{x}^k\}$, where $\mathbf{x}^k$ is defined as:

$$\mathbf{x}^k(\alpha_k) = \frac{1}{\alpha_k} \int\limits_{\mathbf{y}\in B(d_k)} \mathbf{y}G(\mathbf{y})\,d\mathbf{y}. \tag{26}$$

If the function G is unimodal and the optima is unique, the next theorem proves that there is a sequence that converges to the maximizer.

Theorem 3.7. *If the conditions in Theorem* 3.6 *hold, and if* $\exists\varepsilon^{(1)} > 0$, *only* $\mathbf{x}^*$ *satisfies* $\frac{\partial G}{\partial x_i}(\mathbf{x}) = 0$, $i = 1,\ldots,n$, *in* $U(\mathbf{x}^*, \varepsilon^{(1)})$, *then* $\exists\varepsilon^{(2)} > 0$, *any initial point* $\mathbf{x}^0$, *where* $\mathbf{x}^0 \in G^{-1}(B(\mathbf{x}^*, \epsilon^{(2)}))$, *for any* $\varepsilon \in (0, \varepsilon^{(2)}]$, *we can find a sequence* $\{\alpha_k\}$ *s.t. the sequence* $\{\mathbf{x}^k(\alpha_k)\}$ *converges to* $\mathbf{x}^*$.

Proof. Since G is unimodal, therefore $\exists\varepsilon^{(3)} > 0$, s.t. $m_U(\varepsilon) \geqslant M_{E/U}(\varepsilon)$, when $\varepsilon < \varepsilon^{(3)}$. Let $\bar{\varepsilon} = \min\{\varepsilon^{(1)}, \varepsilon^{(2)}, \varepsilon^{(3)}, \varepsilon^*\}$, where ε^* is defined in Theorem 3.6.

Note that $M_{E/U}(\varepsilon)$ is non-increasing with respect to ε if G is unimodal. It then follows that $\exists K > 0$, s.t. $\frac{1}{K} < \varepsilon$, where $\varepsilon \leqslant \bar{\varepsilon}$. By Theorem 3.6, we know that, for any given $x^0 \in G^{-1}(B(\mathbf{x}^*, \varepsilon))$, $\varepsilon \in (0, \bar{\varepsilon}]$, there $\exists\alpha^{(1)} > 0$ and $\exists k_1 > 0$, s.t. when $k > k_1$, $\mathbf{x}^k(\alpha^{(1)}) \geqslant m_U(\frac{1}{K})$. Since $M_{E/U}(\frac{1}{K+1}) \leqslant M_{E/U}(\frac{1}{K}) \leqslant m_U(\frac{1}{K})$, therefore $x^{k_1} \in G^{-1}(B(\mathbf{x}^*, \frac{1}{K}))$. It follows that $\exists\alpha^{(2)} > 0$ and $\exists k_2 > 0$, s.t. when $k > k_2 + k_1$, $\mathbf{x}^k(\alpha^{(2)}) \geqslant m_U(\frac{1}{K+1})$. In general, it follows that there exists $\alpha^{(m)} > 0$ and $\exists k_m > 0$, s.t. when $k > k_m + k_{m-1} + \cdots + k_1$, $\mathbf{x}^k(\alpha^{(m)}) \geqslant m_U(\frac{1}{K+m-1})$.

Set $\alpha_k = \alpha^{(i)}$, when $k_i < k \leqslant k_{i+1}$. Consequently, we can define the sequence $\{x^k(\alpha_k)\}$. Let $\Psi_m = \{G^{-1}(\bar{A}(\mathbf{x}^*, \frac{1}{K+m}))\}$, $m = 1, 2, \ldots$. Since $G^{-1}(A(\mathbf{x}^*, \frac{1}{K})) \supset G^{-1}(A(\mathbf{x}^*, \frac{1}{K+1})) \supset \cdots \supset G^{-1}(A(\mathbf{x}^*, \frac{1}{K+m})) \supset \cdots$, $\lim_{m\to\infty} F(\frac{1}{K+m-1}) = G(\mathbf{x}^*)$ and G is unimodal and continuous, therefore $\bigcap \Psi_m = \{x^*\}$. It then follows that $\lim_{k\to\infty} x^k = x^*$. □

Finally, we study the relationship of the proposed derivative-free algorithm and a general gradient algorithm. We obtain the following theorem.

Theorem 3.8. *If the second partial derivative of* $G(\mathbf{x})$ *exists, then for any given* $\alpha > 0$, *we have*

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \frac{n}{n+2}\frac{\nabla G(\mathbf{x}^k)}{G(\mathbf{x}^k)}\,d_k^2 + O\big(d_k^3\big), \tag{27}$$

where d_k *satisfies* $\int_{B(\mathbf{x}^k,d)} G(\mathbf{x}^k)\,d\mathbf{x} = \alpha$.

Proof. To simplify the notation, let $x_0 = x^k$ and $\mathbf{x}^{k+1} = T(\mathbf{x}^k)$ where T is the operator defined by

$$T\big(\mathbf{x}^k\big) = \frac{1}{\alpha} \int\limits_{B(\mathbf{x}^k,d)} \mathbf{y}G(\mathbf{y})\,d\mathbf{y}. \tag{28}$$

Let $\mathbf{y} = \mathbf{x}_0 + \mathbf{u}$. It then follows that

$$\begin{aligned} T(\mathbf{x}_0) &= \frac{1}{\alpha}\left(\int_B (\mathbf{x}_0 + \mathbf{u})G(\mathbf{x}_0 + \mathbf{u})\,d\mathbf{u}\right) \\ &= \frac{1}{\alpha}\left(\mathbf{x}_0 \int_B G(\mathbf{x}_0 + \mathbf{u})\,du + \int_B \mathbf{u}G(\mathbf{x}_0 + \mathbf{u})\,d\mathbf{u}\right) = \mathbf{x}_0 + \frac{1}{\alpha}M(\mathbf{x}_0), \end{aligned}$$

Please cite this article as: X. Wang et al., A derivative-free optimization algorithm based on conditional moments, J. Math. Anal. Appl. (2006), doi:10.1016/j.jmaa.2006.08.091

where

$$T(\mathbf{x}_0) = \int_B \mathbf{u} G(\mathbf{x}_0 + \mathbf{u})\, du, \tag{29}$$

and $B = \{\mathbf{u} \in R^n\colon \|\mathbf{u}\| \leqslant d\}$.

Expand the density function as follows:

$$G(\mathbf{x}_0 + \mathbf{u}) = G(\mathbf{x}_0) + \mathbf{u}\nabla G(\mathbf{x}_0) + \|\mathbf{u}\|^2 Q(\mathbf{x}_0 + t\mathbf{u}), \quad 0 < t < 1,$$

where

$$Q(\mathbf{x}_0 + t\mathbf{u}) = \frac{1}{2!}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \frac{\partial^2 f}{\partial x_i \partial x_j}\bigg|_{\mathbf{x}=\mathbf{x}_0+t\mathbf{u}}. \tag{30}$$

By using polar coordinates, it can be verified that

$$\int_B d\mathbf{u} = \frac{d^n}{n} C_n,$$

$$\int_B \mathbf{u}\, du = 0,$$

$$\int_B \|\mathbf{u}\|^2\, du = \frac{d^{n+2}}{n+2} C_n,$$

where $C_n = \frac{(\sqrt{\pi})^{n-2}}{\Gamma(\frac{n}{2})} 2\pi$.

We then have

$$\begin{aligned}\alpha &= \int_B G(\mathbf{x}_0 + \mathbf{u})\, d\mathbf{u} \\ &= \int_B G(\mathbf{x}_0)\, du + \int_B \mathbf{u}\nabla G(\mathbf{x}_0)\, d\mathbf{u} + \int_B \|\mathbf{u}\|^2 Q(\mathbf{x}_0 + t\mathbf{u})\, d\mathbf{u} \\ &= \alpha_0 + \alpha_1 + \alpha_2,\end{aligned}$$

where

$$\alpha_0 = G(\mathbf{x}_0)\frac{d^n}{n} C_n,$$

$$\alpha_1 = \nabla G(\mathbf{x}_0) \int_B \mathbf{u}\, dy = 0,$$

$$\alpha_2 = \int_B \|\mathbf{u}\|^2 Q(\mathbf{x}_0 + t\mathbf{u})\, d\mathbf{u} \leqslant M_B \frac{d^{n+2}}{n+2} C_n,$$

where $M_B = \max_B \frac{\partial^2 G(x_1, x_2, \ldots, x_n)}{\partial x_i \partial x_j}$ for any i, j.

This implies that

$$\alpha = G(\mathbf{x}_0)\frac{d^n}{n} C_n + O\big(d^{n+2}\big). \tag{31}$$

Please cite this article as: X. Wang et al., A derivative-free optimization algorithm based on conditional moments, J. Math. Anal. Appl. (2006), doi:10.1016/j.jmaa.2006.08.091

Similarly, we obtain that

$$M(\mathbf{x}_0) = I_0 + I_1 + I_2, \tag{32}$$

where

$$I_0 = G(x_0) \int_B \mathbf{u}\, du = 0,$$

$$I_1 = \nabla G(\mathbf{x}_0) \int_B \|\mathbf{u}\|^2\, du = \nabla G(\mathbf{x}_0) \frac{d^{n+2}}{n+2} C_n,$$

$$I_2 = \int_B \mathbf{u}\|\mathbf{u}\|^2 Q(\mathbf{x}_0 + t\mathbf{u})\, d\mathbf{u} = O\big(d^{n+3}\big).$$

It then follows that

$$M(\mathbf{x}_0) = \nabla G(\mathbf{x}_0) \frac{d^{n+2}}{n+2} C_n + O\big(d^{n+3}\big).$$

We then have

$$\frac{1}{\alpha} M(\mathbf{x}_0) = \frac{n}{n+2} \frac{\nabla G(\mathbf{x}_0)}{G(\mathbf{x}_0)} d^2 + O\big(d^3\big).$$

Moreover, if we ignore the higher order terms in Eq. (27), we then have

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \frac{n}{n+2} \frac{\nabla G(\mathbf{x}^k)}{G(\mathbf{x}^k)} d_k^2. \qquad \square \tag{33}$$

If the limit of the sequence $\mathbf{x}^k$ exists, say $\mathbf{x}^*$, by taking the limit on both sides of the above equation, we then have

$$\mathbf{x}^* = \mathbf{x}^* + \frac{n}{n+2} \frac{\nabla G(\mathbf{x}^*)}{G(\mathbf{x}^*)} d_k^2.$$

It then implies that $\nabla G(\mathbf{x}^*) = 0$. Therefore, this algorithm is also trying to find saddle points whose first-order derivative equal to zero.

Although the first-order derivative is not used in the proposed algorithm, the moving direction of the algorithm is actually closely related to the gradient. For simplicity, let us consider the 1-dimensional case. If $f'(\mathbf{x}^k) > 0$, then the second term in Eq. (33) will be positive. This implies that the $\mathbf{x}^{k+1}$ is generated to the right of $\mathbf{x}^k$ which moves to a "higher" ground. Otherwise, $\mathbf{x}^{k+1}$ will move to the left of $\mathbf{x}^k$. Therefore, the first-order derivative actually dictates the direction of the next move. We emphasize that the first-order derivative is never calculated.

Although we have established convergence properties in this section, we remark that convergence to a local optimum is still possible in our algorithm. For example, for a very small initial value of α, our algorithm could converge to a local optimum. However, we will show through examples in the next section that the outcome of our algorithm is not very sensitive to the choice of the initial values.

4. Numerical experiments

In this section, we will present results of numerical experiments and compare our algorithm with three widely used algorithms: *the Newton*, *the Quasi-Newton* and *the wedge trust region* methods.

Please cite this article as: X. Wang et al., A derivative-free optimization algorithm based on conditional moments, J. Math. Anal. Appl. (2006), doi:10.1016/j.jmaa.2006.08.091

4.1. Comparison with Newton and Quasi-Newton method

The first objective function that we consider is the following:

$$G_{11}(x) = e^{-x^2} + 0.7 * e^{-(x-2)^2} + 10, \quad -\infty < x < \infty. \tag{34}$$

This function is asymmetric and has one global maximum at 0.18 but with an additional local optima around 2. It is mainly flat on most of its domain but has a steep increase near the global and local optima. The function is displayed in the left panel of Fig. 2.

Table 1 provides a comparison between the Newton method and our algorithm. It can be seen that Newton method is very sensitive to the initial values. The Newton method could generate correct results when the initial values are set within the interval $(-0.5, 0.5)$. For other initial values, however, the Newton method does not work well. For example, it is interesting to exam the result generated by the Newton method when the initial value is 1. The location of this particular initial value is actually closer to the global optimum located at 0.18 than the local optimum at 2. However, the Newton method converges to the local saddle point 1.18 instead. If the initial values are set to be on the other side of the local optimum, then the Newton algorithm will be trapped at the local optimum. Furthermore, if the initial values are set to be either -1 or 3 which are both relatively far from the optimum, then the Newton method could not calculate the derivatives at those locations and consequently failed to find either the global or local optima. Among all the initial values we tried, the Newton method could only find the global optima when the initial values are very close to the true global maxima. Another commonly used optimization algorithm for one-dimensional case is the *Secant* method. The Secant method does not require the calculation of the derivative. But it does require the input of 2 parameters. Table 1 also provides the results using different initial values for the Secant method. It can be seen that the Secant method also suffers the similar drawback as the Newton method. First of all, it could also diverge as the Newton method does. Secondly, the accuracy of the algorithm depends heavily on the initial choice of the parameters.

In contrast with the results generated by the Newton method, our algorithm based on conditional moments (CM) can find the global optima regardless of the relative position of the initial value to the global optima. In particular, our algorithm gives a very surprising performance when the initial value is set to be 3. In this case, the starting point lies on the right-hand side of both the global and the local optima. Our algorithm actually jumped across the local optimum and indeed reached the global optimum.

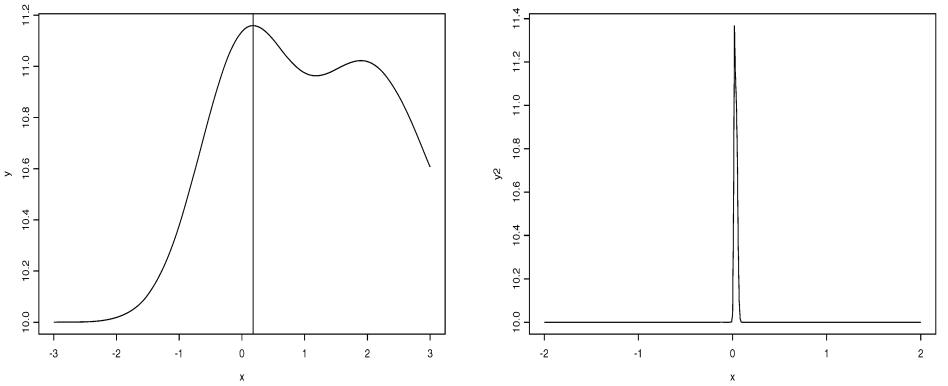

Fig. 2. Objective functions: $e^{-x^2} + 0.7 * e^{-(x-2)^2} + 10$ (left) and $e^{-(50*x-2)^2} + e^{-(100*x-2)^2} + 10$.

Please cite this article as: X. Wang et al., A derivative-free optimization algorithm based on conditional moments, J. Math. Anal. Appl. (2006), doi:10.1016/j.jmaa.2006.08.091

Table 1
Comparison with Newton and Secant method for function G_{11} (tolerance $= 10^{-8}$)

Initials	$x_0 = -1$	$x_0 = -0.5$	$x_0 = 0.0$	$x_0 = 0.5$	$x_0 = 1.0$	$x_0 = 2.5$	$x_0 = 3$
Newton	NA	1.1808	0.1791	0.1791	1.1808	1.8948	NA
CM	0.1806	0.1824	0.1803	0.1812	0.1813	0.1813	0.1813

Initials	$(-1, -0.5)$	$(-1, 0)$	$(-0.5, 0)$	$(-0.5, 1)$	$(0.5, 1.0)$	$(0.5, 1.5)$	$(1, 2)$	$(1, 2.5)$
Secant	NA	0.1791	0.1791	1.1808	1.1808	1.1808	1.8948	NA

Table 2
Comparison with Newton and Secant method for function G_{12} (tolerance $= 10^{-8}$)

	$x_0 = 0$	$x_0 = 0.01$	$x_0 = 0.02$	$x_0 = 0.03$	$x_0 = 0.04$	$x_0 = 0.05$
Newton	NA	NA	0.0229	0.0229	NA	NA
CM	0.0224	0.0224	0.0227	0.0.234	0.0229	0.0229

	$(0, 0.01)$	$(0, 0.03)$	$(0.01, 0.02)$	$(0.01, 0.03)$	$(0.02, 0.04)$	$(0.03, 0.04)$	$(0.03, 0.05)$
Secant	NA	0.0221	0.0221	NA	NA	NA	−0.1282

The second objective function we used is

$$G_{12}(x) = \exp\{-(50x - 2)^2\} + \exp\{-(100x - 2)^2\} + 100, \quad -\infty < x < \infty. \tag{35}$$

This function has one unique optimum at 0.022 but is very steep in its neighbourhood. It is plotted in the right panel in Fig. 2.

The results of the Newton method and ours are presented in Table 2. We see that the dependence on the initial values is quite evident for the Newton method. In fact, the Newton method failed to produce any sensible results unless the initial values are set to be very close to the true optima. The results obtained through the Secant method are also given in Table 2. It can be seen that the Secant method is either divergent or failed to find the true optimum for most initial values we selected. It is highly sensitive to the initial choice of the parameters. For example, the true optimum is found by using initial parameters $(0.01, 0.02)$. However, the Secant method is divergent for a similar pair of parameters, namely $(0.01, 0.03)$.

We now compare the performances of the Newton method and our method for a two-dimensional function:

$$G_{21}(x, y) = e^{-(x^2+y^2)} + e^{-(x-1)-(y-1)^2} + 10, \quad -\infty < x < \infty. \tag{36}$$

The above function has one unique optimum at $(0.5, 0.5)$. Unlike the functions studied in the one-dimensional case, this function is rather flat near the optimum as seen in Fig. 3.

In the two-dimensional case, Quasi-Newton method is also commonly used. We present the results from the Newton, Quasi-Newton and our methods (CM) in Table 3. It is clear that the Newton method is highly sensitive to the initial value of (X_0, Y_0). In fact, the Newton method failed to perform any iteration in many initial values since the method could not calculate the derivatives of the objective function. In comparison, the Quasi-Newton method is only marginally better. Although it delivers the optimum in two cases in which the Newton method failed, the reported values are both grossly far from the true optimum. The Quasi-Newton method, however, would diverge for many initial values. In comparison, our method finds the true optima for most of the initial values. When the initial values are both negative, however, our method reports that the optimum is around $(0, 0)$. Although it seems that our method also missed the target for these cases, carefully examination of the objective function especially the contour plot in Fig. 3 reveals that the optimal value returned by our method is about 11 while the true optimal value

Please cite this article as: X. Wang et al., A derivative-free optimization algorithm based on conditional moments, J. Math. Anal. Appl. (2006), doi:10.1016/j.jmaa.2006.08.091

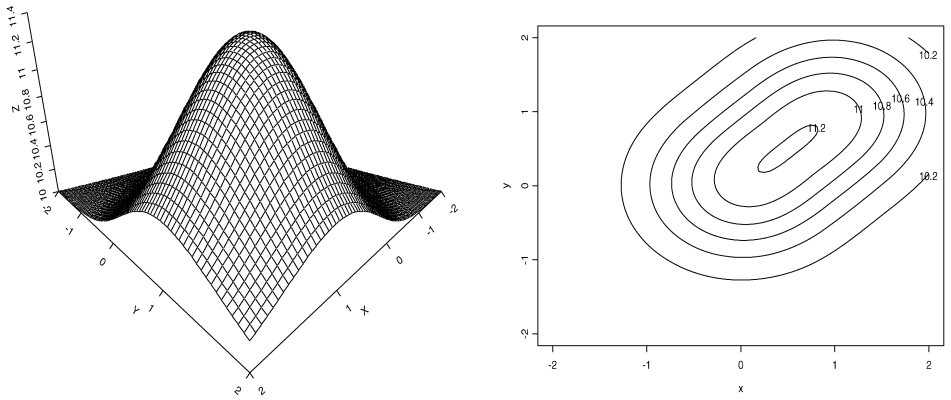

Fig. 3. Surface and contour plot for the function G_{21}.

Table 3
Comparison with Newton and Secant method for function G_{21} (tolerance $= 10^{-8}$)

(X_0, Y_0)	Newton	Quasi-Newton	CM
$(-1.0, -1.0)$	NA	NA	$(-0.00716, -0.00725)$
$(-0.5, -0.5)$	NA	$(-4.0998, -4.0998)$	$(-0.00347, -0.00381)$
$(0.6, 0.6)$	$(0.50000, 0.50000)$	$(0.5000, 0.50000)$	$(0.50001, 0.50001)$
$(1.0, 1.0)$	$(0.50000, 0.50000)$	$(0.50000, 0.50000)$	$(0.50000, 0.50000)$
$(1.2, 1.2)$	$(0.50000, 0.50000)$	$(0.5000, 0.50000)$	$(0.50001, 0.50001)$
$(1.5, 1.5)$	NA	$(5.09988, 5.09988)$	$(0.50000, 0.50000)$
$(2.0, 2.0)$	NA	NA	$(0.50000, 0.50000)$
$(3.0, 3.0)$	NA	NA	$(0.50000, 0.50000)$

for the function is 11.2. Our method could not proceed further since the top of the 2-dimensional function is very flat indeed. Thus those output results by our method seem to be quite reasonable given the nature of the function near the global optimum.

4.2. Comparison with wedge trust region method

We now compare our method with the wedge trust region method (see Marazzi and Nocedal [12]). The wedge trust region method has been shown to be very efficient and accurate for a variety of functions. We indeed apply the wedge trust region method to the functions studied in previous section. The wedge trust region works almost perfectly for those functions with very large or very small first-order derivatives. We proceed to make further comparison in much more complicated situations in which the global optima of the objective functions are accompanied by some local optima.

The first objective function we chose is the function

$$G_{22}(x, y) = 30\exp\big(0.01 * \big[-30(x-2)^2 - 20(y-2)^2\big]\big) + 20\exp\big(-20(x-4)^2 - 5(y-4)^2\big) + 10.$$

The surface and contour plots are provided in Fig. 4. The global optimum is at $(2, 2)$ with a local optimum located at $(4, 4)$. We apply both the wedge trust region method and our method to this function. The results are presented in Table 4. We actually tried many initial values and many of those pairs give almost identical results for both methods. Thus, we only present results that

Please cite this article as: X. Wang et al., A derivative-free optimization algorithm based on conditional moments, J. Math. Anal. Appl. (2006), doi:10.1016/j.jmaa.2006.08.091

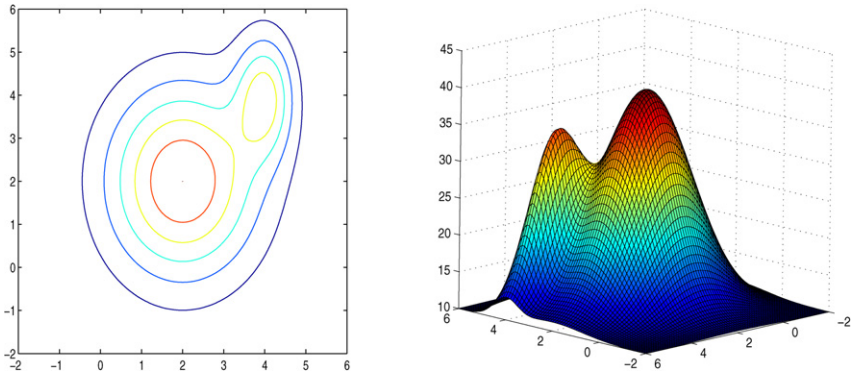

Fig. 4. Comparison with wedge trust method using G_{22}.

Table 4
Comparison between the wedge trust region method and the conditional moment method using G_{22}

Initial values	Wedge trust region	Conditional moment
$(4, 1)$	$(2.000405, 2)$	$(2.004554, 2.003649)$
$(1, 5)$	$(2.000405, 2)$	$(2.005478, 2.004500)$
$(1, -1)$	$(2.000405, 2)$	$(2.006220, 2.005184)$
$(1, 1)$	$(2.000405, 2)$	$(2.000781, 2.000408)$
$(0, 4)$	$(2.000405, 2)$	$(2.005800, 2.004797)$
$(5, 5)$	$(3.923584, 4)$	$(2.001318, 2.000835)$
$(4, 6)$	$(3.923584, 4)$	$(2.001557, 2.001029)$
$(3, 5)$	$(3.923584, 4)$	$(2.002954, 2.002219)$
$(5, 3)$	$(3.923584, 4)$	$(2.002939, 2.002206)$

are representative. The class of top four cases presented in Table 4 demonstrates that these two methods could both find the global optimum for this set of different starting points. If the location of the initial value is not close to the local optimum located at $(4, 4)$, the wedge trust region method is very effective and accurate. In fact, it is more efficient and accurate than our method. For example, if the starting point is chosen at $(1, 1)$ or $(0, 4)$, the wedge trust region locates the true global optimum very accurately while the CM method only converges to the neighbourhood of the location $(2.005, 2.005)$. However, if the initial values are chosen to be close to the local optimum at $(4, 4)$ such as those chosen in the last 4 cases in Table 4, the wedge trust region method will converge to the local optimum instead of the global one. Our CM method, however, converges successfully to the global optimum and ignored the attraction from the local optimum. In summary, the performance of the wedge trust region method also depends on the choice of the initial starting point. Our method, on the other hand, does not rely on the initial value and seem to be more robust although it could be less accurate than the wedge trust region method for some cases.

To further verify our observation that the wedge trust region method might be trapped in a local optimum which is close to the global one, we consider the following function:

$$G_{23}(x) = 30\exp\bigl(0.01 * \bigl[-30(x-2)^2 - 20(y-2)^2\bigr]\bigr) + 20\exp\bigl(-20(x-4)^2 - 5(y-4)^2\bigr).$$

Please cite this article as: X. Wang et al., A derivative-free optimization algorithm based on conditional moments, J. Math. Anal. Appl. (2006), doi:10.1016/j.jmaa.2006.08.091

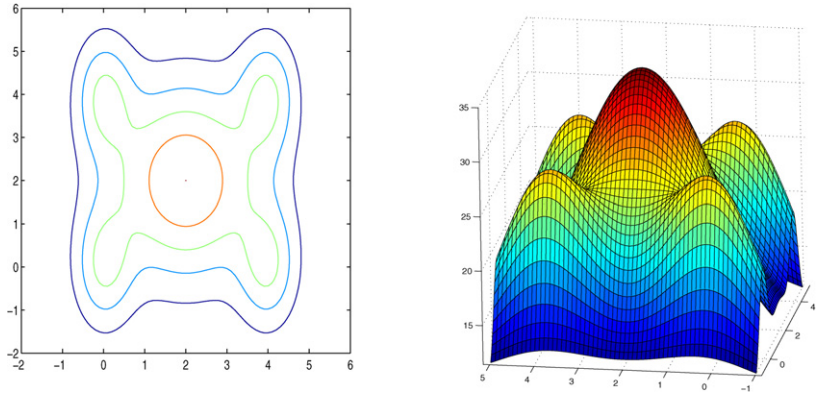

Fig. 5. Surface and contour plot of G_{23}.

Table 5
Comparison between the wedge trust region method and the conditional moment method using function G_{23}

Initial values	Wedge trust region	Conditional moment
$(2, 5)$	$(2.000000, 2.0)$	$(2.000000, 2.000003)$
$(2, -3)$	$(2.000000, 2.0)$	$(2.000000, 1.999999)$
$(-2, 2)$	$(2.000000, 2.0)$	$(2.000001, 2.000000)$
$(2, -1)$	$(2.000000, 2.0)$	$(2.000000, 1.999998)$
$(5, -1)$	$(3.902682, 0.2)$	$(2.000001, 1.999999)$
$(5, 5)$	$(3.902682, 4.0)$	$(2.000000, 2.000004)$
$(-1, -2)$	$(0.009731, 0.2)$	$(2.000000, 1.999974)$
$(-1, 5)$	$(0.009731, 4.0)$	$(1.999999, 2.000001)$
$(5, 2)$	$(3.902682, 4.0)$	$(2.000001, 2.000000)$

The surface and contour plots are given in Fig. 5. As we can see that the global optimum located at $(2, 2)$ is surrounded by four local optima. This is a more challenging case as the objective function changes more radically around the global optimum. We see that both methods would converge to the global optimum if the starting points are in one of the four "valleys." This is not surprising as there exists a clear direct path to the optimum from those locations. However, the wedge trust region method would converge to one of the four local optima if the initial values are close to one of those local optima. In comparison, the CM method could jump over the local optimum nearby and go directly to the global optimum.

We also tried to compare our algorithm with the trust region algorithm for a very challenging function

$$G_{24} = 10^3 \sin(r)/r, \tag{37}$$

where $r = \sqrt{x^2 + y^2}$ and $-10 < x < 10$, $-10 < y < 10$.

The optimum is achieved at the $(0, 0)$ which is the location of singularity of the derivative. Around the global optimum, there is also a ring of infinite local multi-optima within the range of $[-10, 10] \times [-10, 10]$. The graphical presentation of this function is shown in Fig. 6. The left panel shows the 3-D picture of the function with the singularity removed. The sectional plot of the function for $y = 0$ is provided in the right panel of Fig. 6. Please note that the optimum has

Please cite this article as: X. Wang et al., A derivative-free optimization algorithm based on conditional moments, J. Math. Anal. Appl. (2006), doi:10.1016/j.jmaa.2006.08.091

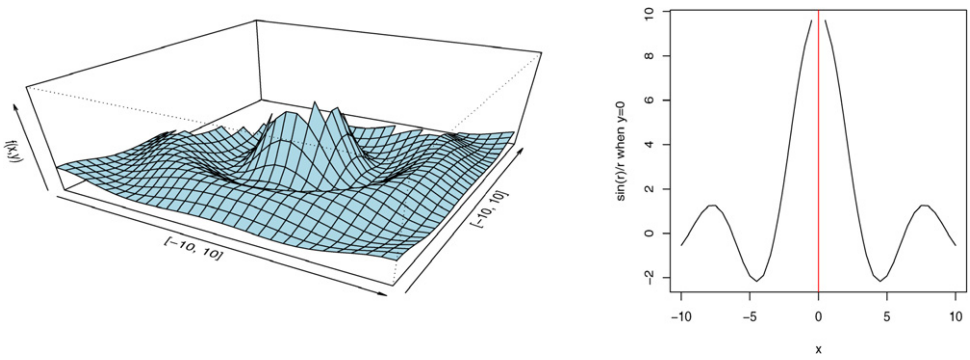

Fig. 6. Surface and section plots of function $G_{24}(x, y)/100$.

Table 6
Comparison between the wedge trust region method and the conditional moment method using function G_{24}

Initial values	Wedge trust region	Conditional moment
(4.0, 5.0)	(5.85, 5.00)	(0.02, 0.06)
(5.0, 5.0)	(5.32, 6.00)	(−0.01, −0.01)
(5.0, 6.0)	(5.04, 6.00)	(−0.05, 0.04)
(6.0, 7.0)	(4.65, 6.00)	(−0.06, 0.03)
(7.0, 7.0)	(5.98, 5.00)	(0.03, 0.07)

been removed in order to plot the graph. The numerical results are given in Table 6. Since the function is symmetric, we only provide results using initial points from the first quadrant. The results using initial values from other areas are very similar. It can be seen from Table 6 that the trust region method could not even get close to the global optimum and is trapped by the ring of infinitely many surrounding local optima. Our method, however, can break through the collection of infinite multi-optima and get reasonably close to the global optima. Due to the singularity of the derivative at the global optimum, our method could not get very close to the true optima for one run. The precision could be improved iteratively by applying a strictly decreasing sequence of α^k by using the previous stopping point as the new starting point. However, the trust region method would not get any better result.

5. Conclusion

In this article we propose a derivative-free algorithm for optimization. The novel feature of the algorithm lies in the fact that it is based on conditional moments calculated from local integrations and does not require the evaluations or knowledge of any order derivatives of the objective function. The local integrations can be evaluated by the numerical quadratures for each dimension separately to avoid integration on high-dimensional space. The parameters d_k and α_k in the algorithm are calculated adaptively by Newton type of iterations. We also provide theoretical analysis to provide insights on the proposed algorithm. Numerical results based on various one-dimensional and two-dimensional functions have shown that the algorithm could be very effective and accurate when compared with those widely used optimization algorithms.

Please cite this article as: X. Wang et al., A derivative-free optimization algorithm based on conditional moments, J. Math. Anal. Appl. (2006), doi:10.1016/j.jmaa.2006.08.091

References

[1] L. Breiman, Probability, Classics Appl. Math., SIAM, 1992.
[2] C.G. Broyden, The convergence of a class of double-rank minimization algorithms 2, The new algorithm, J. Inst. Math. Appl. 6 (1970) 222–231.
[3] R.L. Burden, J.D. Faires, Numerical Analysis, eighth ed., Brooks/Cole, 2005.
[4] J. Ding, T.Y. Li, A polynomial-time predictor–corrector algorithm for a linear complementary, SIAM J. Optim. 1 (1991) 83–92.
[5] W.C. Davidon, Variable Metric Method for Minimization, A.E.C. Research and Development Report ANL-5590, Argonne National Laboratory, Chicago, 1959.
[6] R. Fletcher, A new approach to variable-metric algorithms, Computer J. 13 (1970) 317–322.
[7] P.E. Frandsen, K. Jonasson, H.B. Nielsen, O. Tingleff, Unconstrained Optimization, IMM, 2004.
[8] D. Goldfarb, A family of variable-metric algorithms derived by variational means, Math. Comp. 24 (1970) 23–26.
[9] W.J. Kennedy, J.E. Gentel, Statistical Computing, Marcel Dekker, 1980.
[10] V. Kreinovich, Probability, intervals, what nest? Optimization problems related to extension of interval computations to situations with partial information about probability, J. Global Optim. 29 (2004) 265–280.
[11] H.M. Kvamsdal, H.F. Svendsen, T. Hertzberg, O. Olsvik, Dynamic simulation and optimization of a catalytic steam reformer, Chem. Engrg. Sci. 54 (1999) 2697–2706.
[12] M. Marazzi, J. Nocedal, Wedge Trust Region Methods for Derivative Free Optimization, Springer-Verlag, 2002.
[13] J. Nocedal, S.J. Wright, Numerical Optimization, Springer-Verlag, 1999.
[14] E. Polak, Optimization: Algorithms and Consistent Approximations, Springer, 1997.
[15] M.J.D. Powell, A direct search optimization method that models the objective and constraint function by linear interpolation, in: S. Gomez, J.P. Hennart (Eds.), Advances in Optimization and Numerical Analysis, Kluwer Academic Publ., 1994, pp. 51–67.
[16] C.P. Robert, G. Casella, Monte Carlo Statistical Methods, Springer-Verlag, New York, 2004.
[17] D.F. Shanno, Conditioning of quasi-Newton methods for function minimization, Math. Comp. 24 (1970) 647–656.
[18] J. Stoer, R. Bulirsch, Introduction to Numerical Analysis, Springer-Verlag, 1993.
[19] M.H. Wright, Direct search methods: once scorned not respectable, in: Numerical Analysis, 1995, Proceedings of the 1995 Dundee Biennial Conference in Numerical Analysis, Addison Wesley Longman, 1996, pp. 191–208.
[20] S. Zhang, S. Wang, X. Deng, Portfolio selection theory with different interests rates for borrowing and lending, J. Global Optim. 28 (2004) 67–95.

Please cite this article as: X. Wang et al., A derivative-free optimization algorithm based on conditional moments, J. Math. Anal. Appl. (2006), doi:10.1016/j.jmaa.2006.08.091